TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM KHOA CÔNG NGHỆ THÔNG TIN

MÔN: KIẾN TRÚC MÁY TÍNH VÀ HỢP NGỮ LỚP: 16CNTN

**BÁO CÁO ĐỒ ÁN 1**

BIỂU DIỄN VÀ TÍNH TOÁN SỐ HỌC

TRÊN MÁY TÍNH

**Người thực hiện:**

**Nguyễn Quốc Vương – 1612829**

**Nguyễn Thanh Tuấn – 1612774**

**Lê Thành Công – 1612842**

**Giáo viên:**

**Lý thuyết: Phạm Tuấn Sơn**

**Thực hành: Lê Viết Long**

Mục lục

[I. Nội dung. 1](#_Toc510036380)

[1. Đánh giá mức độ hoàn thành. 1](#_Toc510036381)

[2. Phạm vi biểu diễn của các kiểu dữ liệu thiết kế. 1](#_Toc510036382)

[3. Mô tả tổng quan đồ án. 2](#_Toc510036383)

[a. Số Nguyên 2](#_Toc510036384)

[b. Số chấm động 5](#_Toc510036385)

[4. Hình ảnh giao diện chương trình với các test case. 9](#_Toc510036386)

[c. Ví dụ trường hợp đọc vào số QInt. 9](#_Toc510036387)

[d. Ví dụ trường hợp đọc vào số QFloat. 11](#_Toc510036388)

[5. Tài liệu tham khảo 12](#_Toc510036389)

[II. Đánh giá thành viên 12](#_Toc510036390)

1. **Nội dung.**
2. **Đánh giá mức độ hoàn thành.**

Đánh giá tổng quan: Hoàn thành tất cả các yêu cầu của độ án (100%).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Nội dung** | | **Tỷ lệ hoàn thành (%)** | **Ghi chú** |
| 1 | Số nguyên lớn (QInt) | Hàm nhập | 100 |  |
| 2 | Hàm xuất | 100 |  |
| 3 | Hàm chuyển đổi QInt từ thập phân sang nhị phân | 100 |  |
| 4 | Hàm chuyển đổi QInt từ nhị phân sang thập phân | 100 |  |
| 5 | Hàm chuyển đổi QInt từ nhị phân sang thập lục phân | 100 |  |
| 6 | Hàm chuyển đổi QInt từ thập phân sang thập lục phân | 100 |  |
| 7 | Các toán tử: “+”, “-”, “\*”, “/”, “%” | 100 | Operator “%”: Làm thêm |
| 8 | Các toán tử: AND “&”, OR “|”, XOR “^”, NOT “~” | 100 |  |
| 9 | Các toán tử: dịch trái “<<”, dịch phải “>>” | 100 |  |
| 10 | Số chấm động chính xác cao (QFloat) | Hàm nhập | 100 |  |
| 11 | Hàm xuất | 100 |  |
| 12 | Hàm chuyển đổi QFloat nhị phân sang thập phân | 100 |  |
| 13 | Hàm chuyển đổi QFloat thập phân sang nhị phân | 100 |  |
| 14 | Các operator tính toán: “+”, “-”, “\*”, “/” | 100 |  |

Bảng 1. Bảng đánh giá chi tiết các yêu cầu của đồ án 1

1. **Phạm vi biểu diễn của các kiểu dữ liệu thiết kế.**

* Số nguyên lớn (QInt):

+ Số lớn nhất có thể biểu diễn: 2127-1

+ Số nhỏ nhất có thể biểu diễn: -2127

* Số chấm động chính xác cao (QFloat):

+ Số dương lớn nhất có thể biểu diễn: (2-2-112).216383

+ Số dương nhỏ nhất có thể biểu diễn: 2-16382

+ Số âm lớn nhất có thể biểu diễn: -2-16382

+ Số âm nhỏ nhất có thể biểu diễn: -(2-2-112).216383

1. **Mô tả tổng quan đồ án.**
   1. **Số Nguyên**

* Kiểu dữ liệu biểu diễn: **usigned int data[4]** (Mảng số nguyên không dấu 4 phần tử), tương đương 16 bytes = 128 bits.
* Dữ liệu đầu vào và ra có thể là dãy số nhị phân, số thập phân, số thập lục phân.
* Chuyển đổi:
* Ý tưởng: Dù dữ liệu đầu vào ở dạng nào thì mục đích cuối vẫn là chuyển về chuỗi nhị phân sau đó đưa từng bit lên data. Ta có 2 hàm quan trọng là **void setBit(int value, int pos)** với nhiệm vụ gán giá trị 0 hoặc 1 tại vị trí bất kì trên data và hàm **int getBit(int pos)** để lấy ra giá trị bit tại vị trí bất kì. Với vị trí nằm trong khoảng [0, 127].
* Đối với dãy số nhị phân, đưa trực tiếp từng bit lên data.
* Đối với số thập phân, lấy số cần chuyển đổi chia cho 2 (kết quả chỉ lấy phần nguyên), sau đó tiếp tục lấy kết quả chia 2 (và cũng chỉ lấy phần nguyên), kết quả số nhị phân thu được là tập hợp các số dư của các phép chia.
* Các bước thực hiện:

+ Bước 1: Kiểm tra số âm hay dương.

+ Bước 2:

* Nếu là số dương, chuyển số nguyên dương sang nhị phân.
* Nếu là số âm, chuyển phần dương sang nhị phân, đảo bit sau đó cộng thêm 1 vào kết quả vừa nhận được.

+ Bước 3: Đưa dãy nhị phân kết quả lên data.

* Đối với số thập lục phân, ta chuyển đổi theo quy tắc lấy từ ký tự chuyển sang dạng 4 bit đã biết rồi đưa dãy bit lên data

VD: AB => 1010 1011

* Lấy kết quả:
  + Ý tưởng: Dữ liệu được lưu trên data thực chất là dãy bit. Ta dựa vào là getBit để lấy ra được dãy bit đã lưu và xử lý
* Đối với dãy số nhị phân, ta lấy trực tiếp dãy bit từ data
* Đối với số thập phân, bắt đầu từ bên trái, nhân đôi kết quả, rồi cộng con số bên cạnh cho đến khi không còn con số nào nữa.
* Các bước thực hiện:

+ Bước 1: Kiểm tra dấu của số nhị phân. Nếu là 0 thì là số dương, 1 là số âm

+ Bước 2:

* Nếu là số dương, ta xuống bước 3
* Nếu là số âm, trừ đi 1 của số nhị phân rồi đảo tất cả các bit đã thu được

+ Bước 3: Chuyển bit nhận được sang hệ thập phân. Số thập phân kết quả sẽ là tổng các tích của kí tự nhị phân x 2 lũy thừa vị trí.

* Đối với số thập lục phân, lấy ra từng lô 4 bit rồi tiến hành chuyển ra hệ thập lục phân theo quy tắc
* Tính toán: thực hiện trên dãy bit của data

+ Phép cộng:

0 + 0 = 0

0 + 1 = 1

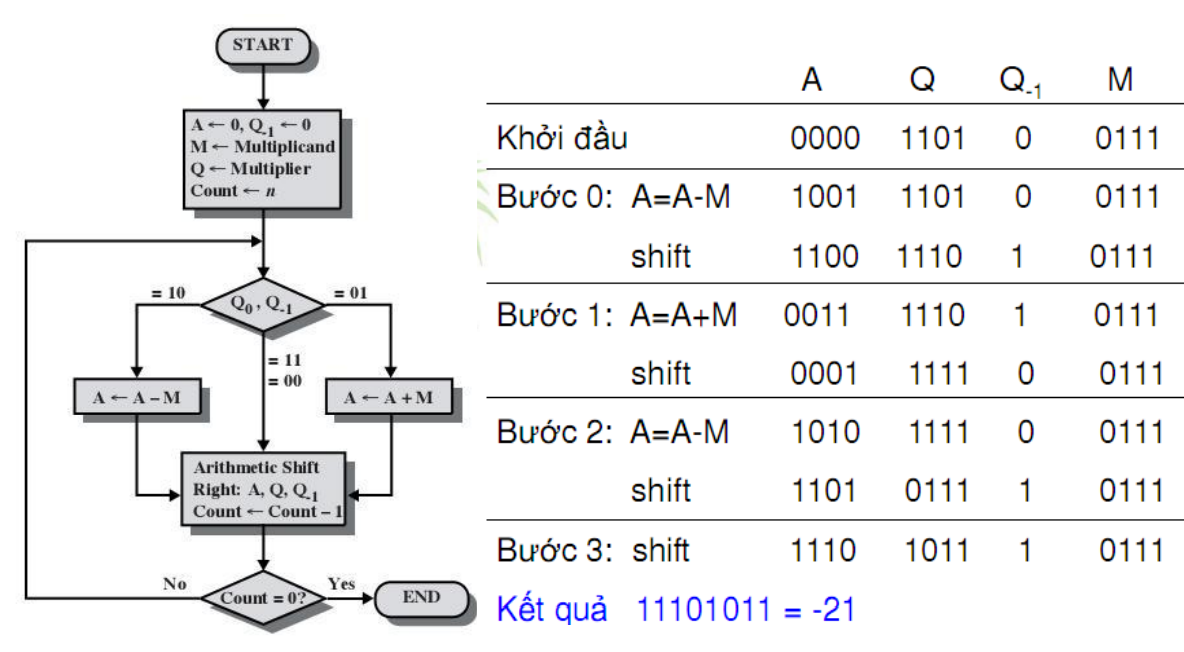
1 + 0 = 1

1 + 1 = 10 (nhớ 1 lên hàng thứ 2)

+ Phép trừ: chuyển số có dấu trừ đứng trước thành số bù 2 (đảo bit rồi cộng

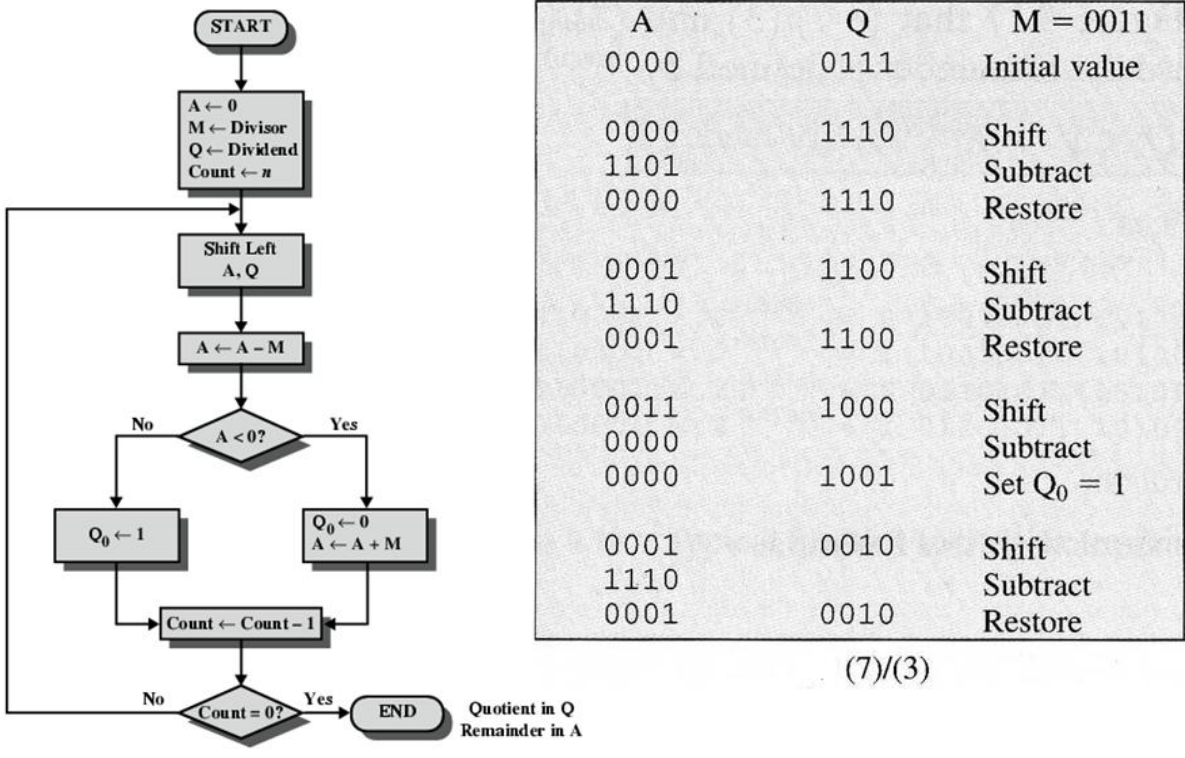
1), sau đó cộng bình thường.

+ Phép nhân: thuật toán Booth.



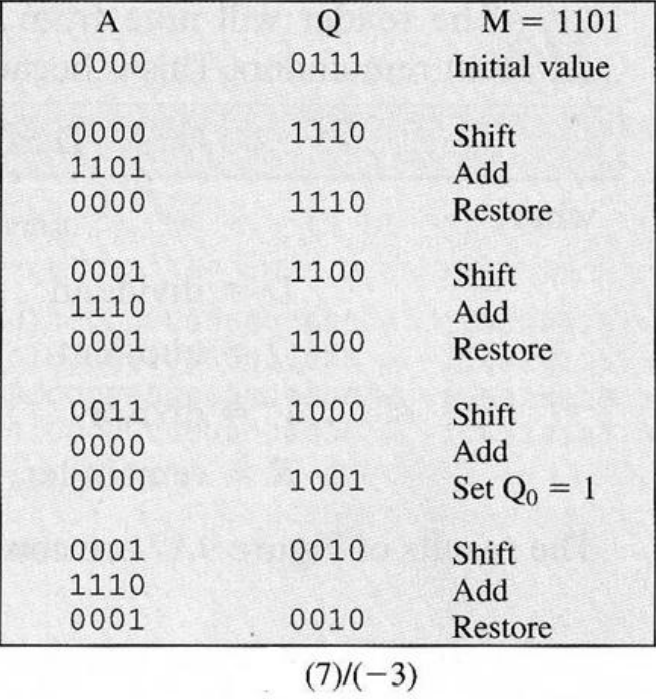
Hình 1. Thuật toán Booth

+ Phép chia: Thuật toán chia không dấu.



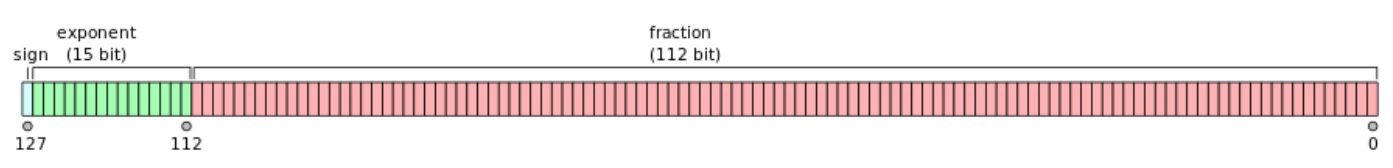
Hình 2. Thuật toán chia nguyên không dấu

* Thuật toán chia có dấu.
* Thực hiện như phép chia không dấu.
* Nếu số chia và số bị chia khác dấu thì đổi dấu thương.



Hình 3. Chia số nguyen có dấu

* 1. **Số chấm động**
* Kiểu dữ liệu biểu diễn: **usigned int data[4]** (mảng số nguyên không dấu 4 phần tử) tương đương 16 bytes = 128 bits.



Hình 4. Phân chia số bit biểu diễn số thực 128 bits (16 bytes)

* Dữ liệu đầu vào và ra có thể là dãy số nhị phân, số thập phân.
* Thuật toán chuyển đổi:
* Chuyển từ giá trị số chấm động sang biểu diễn nhị phân.

1. Chuyển phần trị sang dạng nhị phân.

a. Phần nguyên: theo thuật toán chuyển từ số thập phân sang nhị phân.

b. Phần thập phân: Lặp lại việc nhân phần thập phân với 2 cho tới khi phần thập phân bằng 0 (hoặc đủ số bit phần trị). Tại mỗi bước nhân, sẽ phát sinh 1bit tùy thuộc vào phần nguyên của kết quả phép nhân.

Ví dụ:

0.625 x 2 = 1.25 => 1

0.25 x 2 = 0.5 => 0

0.5 x 2 = 1.0 => 1

Như vậy, ta có được biểu diễn nhị phân của phần trị.

2. Chuẩn hóa.

Ví dụ: 1100.101 = 1.100101 x 23

3. Điền các bit vào các trường theo chuẩn:

- Dấu: số âm nên bit dấu S = 1, ngược lại S = 0.

- Phần mũ: cộng phần mũ của 2 (tìm được sau bước chuẩn hóa) với phần bias (2k-1 - 1).

- Phần trị: thêm bit 0 vào bên phải cho đủ số bit phần trị.

Như vậy, ta có được biểu diễn nhị phân của số chấm động.

* Chuyển từ biểu diễn nhị phân của số chấm động sang biểu diễn thập phân

1. Chia thành 3 thành thành phần.

S = dấu (bit đầu tiên)

E = mũ (15bit tiếp theo)

M = phần trị (112bit còn lại)

2. Tính phần mũ: đổi dãy nhị phân thành số thập phân rồi trừ cho bias

(2k-1 - 1).

3. Tính phần trị: phục hồi bit 1 ở phần nguyên và loại bỏ các bit 0 thừa ở cuối.

4. Kết hơp phần trị với phần mũ:

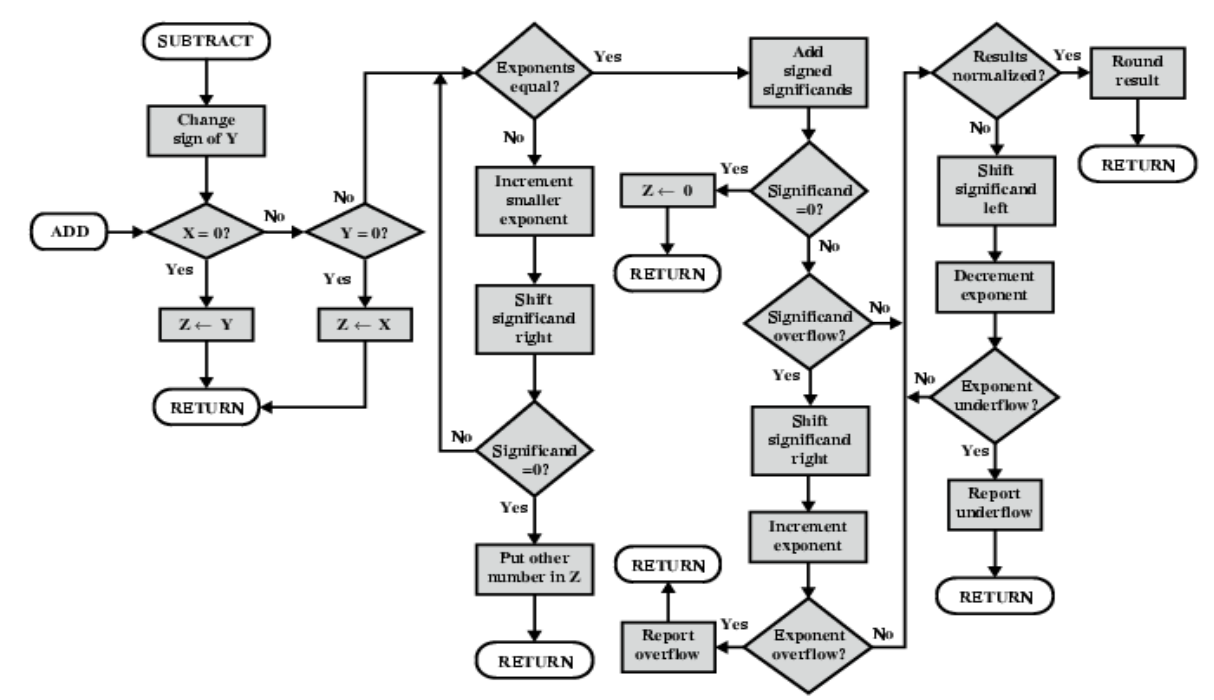
Nếu trị tuyệt đối của phần mũ quá lớn, không thể kết hợp với phần trị như trên, thì có thể thực hiện tính giá trị của phần mũ rồi nhân với phần trị sẽ cho kết quả gần đúng.

5. Chuyển dãy nhị phân thành giá trị số chấm động tương ứng

6. Thêm dấu

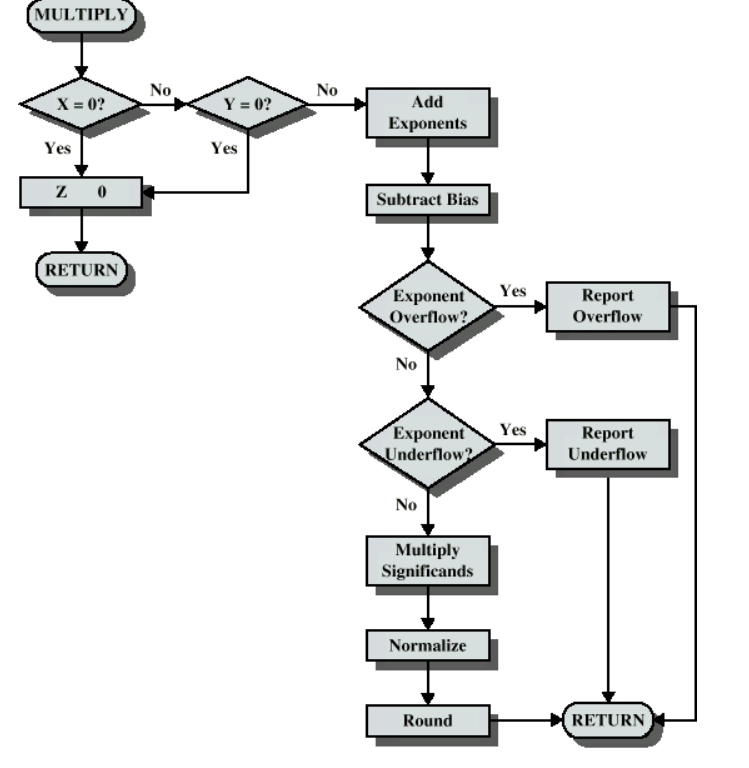
* Tính toán:

+ Phép cộng, trừ:



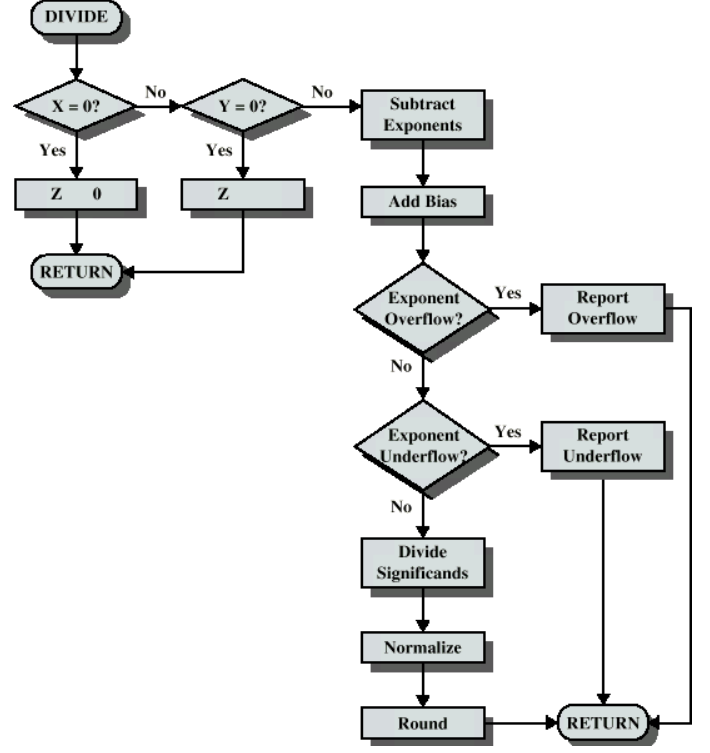
Hình 5. Thuật toán cộng, trừ số thực

+ Phép nhân:



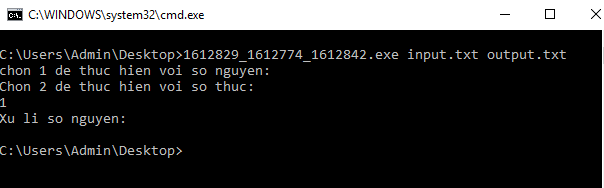
Hình 6. Thuật toán nhân số thực

+ Phép chia:



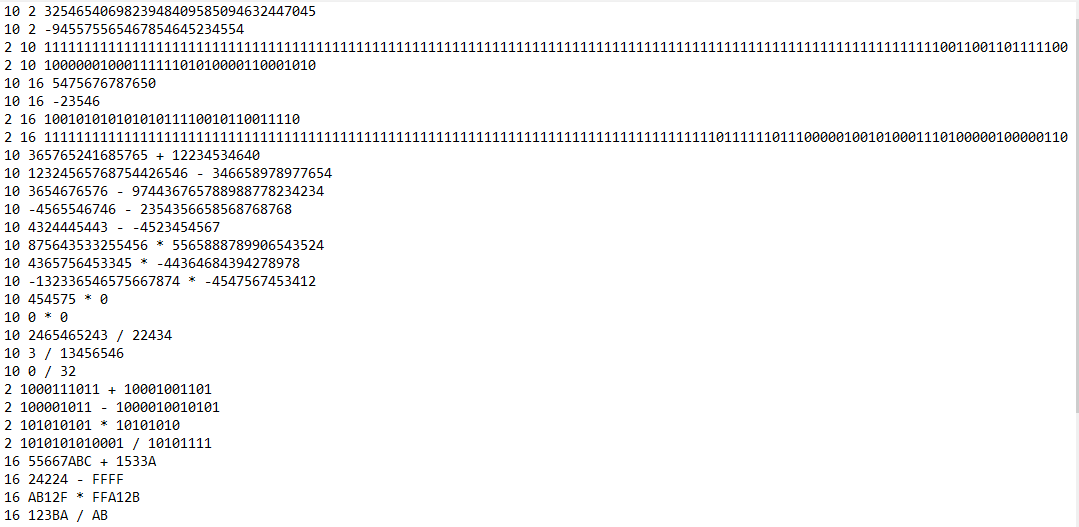
Hình 7. Thuật toán chia số thực

1. **Hình ảnh giao diện chương trình với các test case.**
   1. **Ví dụ trường hợp đọc vào số QInt.**

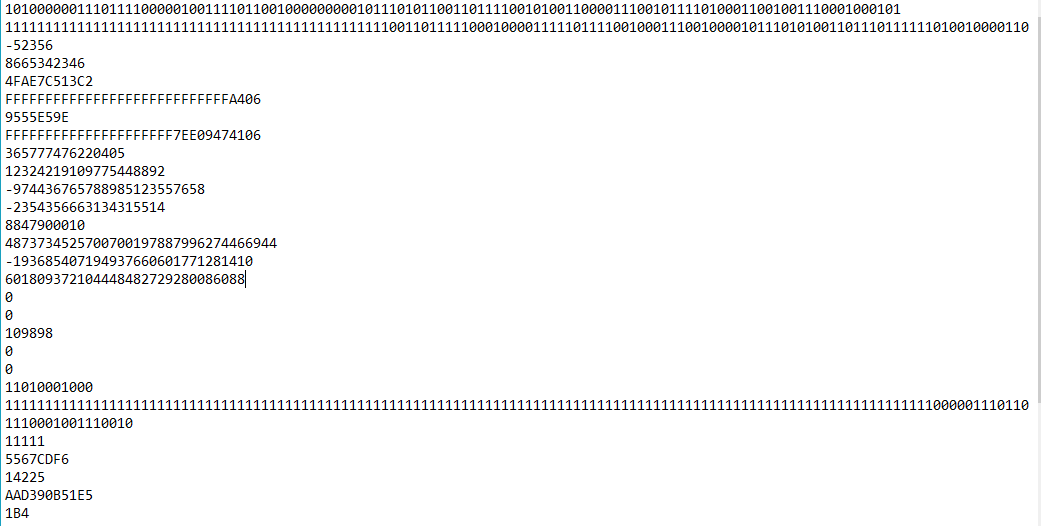


Hình 8. Lệnh thao tác command line

* Test case các phép chuyển đổi và phép toán số học

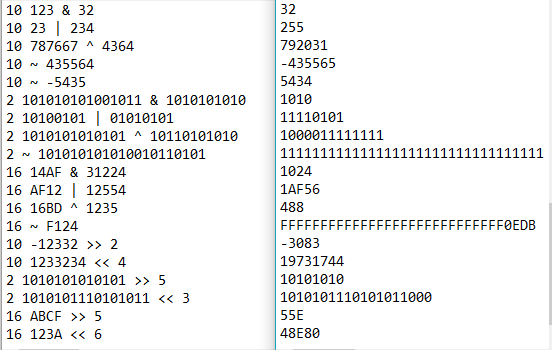


Hình 9. Dữ liệu đầu vào các phép chuyển đổi và phép toán



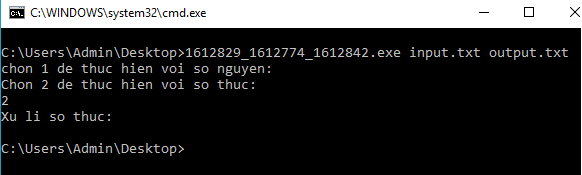
Hình 10. Dữ liệu đầu ra các phép chuyển đổi và phép toán

* Test case toán tử &, |, ^, ~, <<, >>



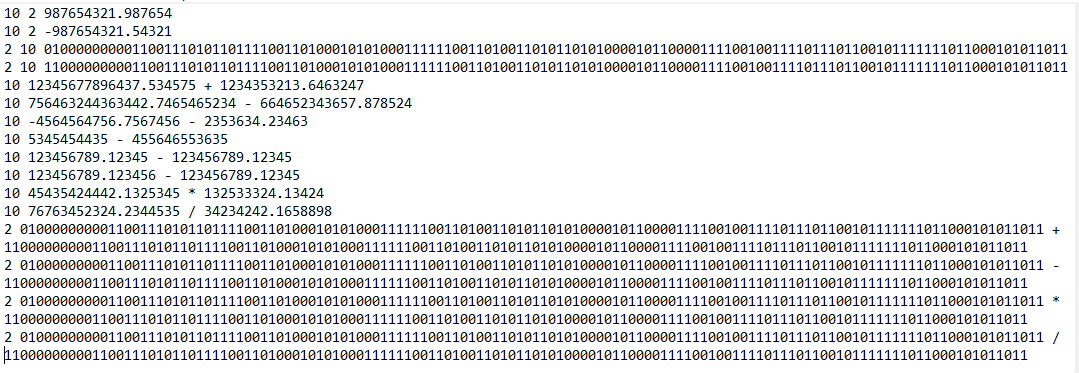
Hình 11. Dữ liệu đầu vào (bên trái) và đầu ra (bên phải) các toán tử &, |, ^, ~, <<, >>

* 1. **Ví dụ trường hợp đọc vào số QFloat.**

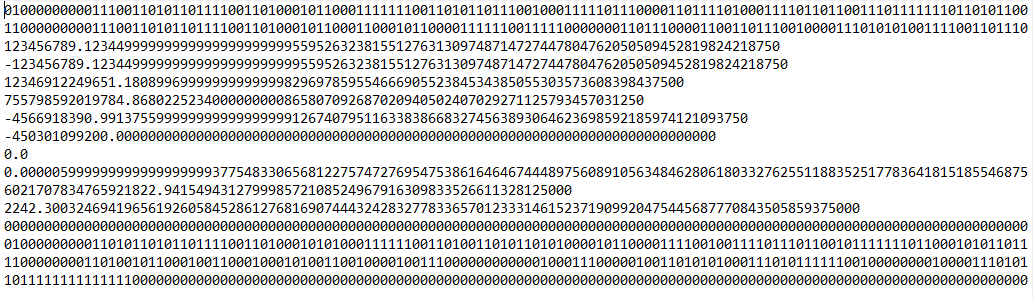


Hình 12. Lệnh thao tác command line

* Test case các phép chuyển đổi và phép toán số học



Hình 13. Dữ liệu đầu vào các thao tác với số thực



Hình 14. Dữ liệu đầu ra các thao tác với số thực

1. **Tài liệu tham khảo**

[1]. Tài liệu giảng dạy “Bai02\_So\_nguyen.pdf”, GV. Phạm Tuấn Sơn, ĐH. KHTN

[2]. Tài liệu giảng dạy “Bai03\_So\_cham\_dong.pdf”, GV. Phạm Tuấn Sơn, ĐH. KHTN

[3].https://vi.wikipedia.org/wiki/S%E1%BB%91\_th%E1%BB%B1c\_d%E1%BA%A5u\_ph%E1%BA%A9y\_%C4%91%E1%BB%99ng

[4]. <http://binary-system.base-conversion.ro/convert-real-numbers-from-decimal-system-to-64bit-double-precision-IEEE754-binary-floating-point.php>

1. **Đánh giá thành viên**

|  |  |  |
| --- | --- | --- |
| **MSSV** | **Họ và tên** | **Đánh giá hoàn thành** |
| 1612829 | Nguyễn Quốc Vương | Tốt |
| 1612774 | Nguyễn Thành Tuấn | Tốt |
| 1612842 | Lê Thành Công | Tốt |